Practice2\_2019

Lukyanova Anastasia PMI IV-1

8 марта 2019 г

library('dplyr')  
library('lattice')  
library('ggplot2')  
library('data.table')

согласно заданию, необходимо построить три графика, каждый из которых выполнен в одной из графических библиотек: ‘base’, ‘laticce’ и ‘ggplot’. На моих графиках будет изображена динамика суммарной массы поставок с сентября по декабрь. Если будут обнаружены пропуски, они будут заменены на медианы. Цветом покажем три группы стран поставщиков: Таможенный союз, СНГ (за исключением стран, входящих в таможенный союз) и остальные страны.

Первым шагом загрузим данные.

# загружаем файл с данными по импорту масла в РФ (из прошлой практики)  
fileURL <- 'https://raw.githubusercontent.com/aksyuk/R-data/master/COMTRADE/040510-Imp-RF-comtrade.csv'  
# создаём директорию для данных, если она ещё не существует:  
if (!file.exists('./data')) {  
 dir.create('./data')  
}  
# создаём файл с логом загрузок, если он ещё не существует:  
if (!file.exists('./data/download.log')) {  
 file.create('./data/download.log')  
}  
# загружаем файл, если он ещё не существует,  
# и делаем запись о загрузке в лог:  
if (!file.exists('./data/040510-Imp-RF-comtrade.csv')) {  
 download.file(fileURL, './data/040510-Imp-RF-comtrade.csv')  
 # сделать запись в лог  
 write(paste('Файл "040510-Imp-RF-comtrade.csv" загружен', Sys.time()),   
 file = './data/download.log', append = T)  
}  
# читаем данные из загруженного .csv во фрейм, если он ещё не существует  
if (!exists('DT')){  
 DT.import <- data.table(read.csv('./data/040510-Imp-RF-comtrade.csv',   
 stringsAsFactors = F))  
}  
# предварительный просмотр  
dim(DT.import) # размерность таблицы

## [1] 416 18

str(DT.import) # структура (характеристики столбцов)

## Classes 'data.table' and 'data.frame': 416 obs. of 18 variables:  
## $ Classification : chr "HS" "HS" "HS" "HS" ...  
## $ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...  
## $ Period : int 201008 201001 201002 201007 201007 201009 201001 201002 201004 201011 ...  
## $ Period.Desc : chr "August 2010" "January 2010" "February 2010" "July 2010" ...  
## $ Aggregate.Level: int 6 6 6 6 6 6 6 6 6 6 ...  
## $ Is.Leaf.Code : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Trade.Flow.Code: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Trade.Flow : chr "Imports" "Imports" "Imports" "Imports" ...  
## $ Reporter.Code : int 51 97 97 97 246 268 276 276 842 842 ...  
## $ Reporter : chr "Armenia" "EU-27" "EU-27" "EU-27" ...  
## $ Partner.Code : int 643 643 643 643 643 643 643 643 643 643 ...  
## $ Partner : chr "Russian Federation" "Russian Federation" "Russian Federation" "Russian Federation" ...  
## $ Commodity.Code : int 40510 40510 40510 40510 40510 40510 40510 40510 40510 40510 ...  
## $ Commodity : chr "Dairy produce; derived from milk, butter" "Dairy produce; derived from milk, butter" "Dairy produce; derived from milk, butter" "Dairy produce; derived from milk, butter" ...  
## $ Qty.Unit.Code : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Netweight.kg : int 200 12 120 57600 38400 10000 NA 100 NA NA ...  
## $ Trade.Value.USD: int 1170 68 502 302318 201551 55926 71 506 3304 4467 ...  
## $ Flag : int 0 0 0 0 0 0 0 0 0 0 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

DT.import # удобный просмотр объекта data.table

## Classification Year Period Period.Desc Aggregate.Level  
## 1: HS 2010 201008 August 2010 6  
## 2: HS 2010 201001 January 2010 6  
## 3: HS 2010 201002 February 2010 6  
## 4: HS 2010 201007 July 2010 6  
## 5: HS 2010 201007 July 2010 6  
## ---   
## 412: HS 2018 201804 April 2018 6  
## 413: HS 2018 201805 May 2018 6  
## 414: HS 2018 201808 August 2018 6  
## 415: HS 2018 201809 September 2018 6  
## 416: HS 2018 201810 October 2018 6  
## Is.Leaf.Code Trade.Flow.Code Trade.Flow Reporter.Code  
## 1: 1 1 Imports 51  
## 2: 1 1 Imports 97  
## 3: 1 1 Imports 97  
## 4: 1 1 Imports 97  
## 5: 1 1 Imports 246  
## ---   
## 412: 1 1 Imports 842  
## 413: 1 1 Imports 842  
## 414: 1 1 Imports 842  
## 415: 1 1 Imports 842  
## 416: 1 1 Imports 842  
## Reporter Partner.Code Partner  
## 1: Armenia 643 Russian Federation  
## 2: EU-27 643 Russian Federation  
## 3: EU-27 643 Russian Federation  
## 4: EU-27 643 Russian Federation  
## 5: Finland 643 Russian Federation  
## ---   
## 412: United States of America 643 Russian Federation  
## 413: United States of America 643 Russian Federation  
## 414: United States of America 643 Russian Federation  
## 415: United States of America 643 Russian Federation  
## 416: United States of America 643 Russian Federation  
## Commodity.Code Commodity Qty.Unit.Code  
## 1: 40510 Dairy produce; derived from milk, butter 0  
## 2: 40510 Dairy produce; derived from milk, butter 0  
## 3: 40510 Dairy produce; derived from milk, butter 0  
## 4: 40510 Dairy produce; derived from milk, butter 0  
## 5: 40510 Dairy produce; derived from milk, butter 0  
## ---   
## 412: 40510 Dairy produce; derived from milk, butter 0  
## 413: 40510 Dairy produce; derived from milk, butter 0  
## 414: 40510 Dairy produce; derived from milk, butter 0  
## 415: 40510 Dairy produce; derived from milk, butter 0  
## 416: 40510 Dairy produce; derived from milk, butter 0  
## Netweight.kg Trade.Value.USD Flag  
## 1: 200 1170 0  
## 2: 12 68 0  
## 3: 120 502 0  
## 4: 57600 302318 0  
## 5: 38400 201551 0  
## ---   
## 412: 1085 10944 0  
## 413: 852 6936 0  
## 414: 414 3962 0  
## 415: 436 4228 0  
## 416: 634 6228 0

Далее получим срез необходимых данных, запишем их в отдельную data.table, проверим на наличие пропусков, в случае их обнаружения, заполним их медианами.

# сколько месяцев в данных?  
unique(DT.import$Period.Desc)

## [1] "August 2010" "January 2010" "February 2010" "July 2010"   
## [5] "September 2010" "April 2010" "November 2010" "October 2011"   
## [9] "November 2011" "December 2011" "March 2011" "July 2011"   
## [13] "January 2011" "February 2011" "April 2011" "June 2011"   
## [17] "August 2011" "September 2011" "May 2011" "May 2012"   
## [21] "June 2012" "October 2012" "November 2012" "August 2012"   
## [25] "January 2012" "February 2012" "March 2012" "April 2012"   
## [29] "July 2012" "September 2012" "December 2012" "January 2013"   
## [33] "February 2013" "March 2013" "April 2013" "May 2013"   
## [37] "June 2013" "July 2013" "August 2013" "September 2013"  
## [41] "October 2013" "November 2013" "December 2013" "November 2014"   
## [45] "January 2014" "August 2014" "February 2014" "March 2014"   
## [49] "April 2014" "May 2014" "June 2014" "July 2014"   
## [53] "September 2014" "October 2014" "December 2014" "February 2015"   
## [57] "March 2015" "November 2015" "December 2015" "January 2015"   
## [61] "April 2015" "May 2015" "June 2015" "July 2015"   
## [65] "August 2015" "September 2015" "October 2015" "February 2016"   
## [69] "March 2016" "April 2016" "May 2016" "July 2016"   
## [73] "August 2016" "September 2016" "October 2016" "November 2016"   
## [77] "December 2016" "June 2016" "January 2016" "January 2017"   
## [81] "February 2017" "March 2017" "April 2017" "May 2017"   
## [85] "June 2017" "July 2017" "August 2017" "September 2017"  
## [89] "October 2017" "November 2017" "December 2017" "January 2018"   
## [93] "February 2018" "March 2018" "April 2018" "May 2018"   
## [97] "July 2018" "August 2018" "June 2018" "September 2018"  
## [101] "October 2018"

#фильтруем данные для получения необходимого среза данных  
(inform <- data.table(filter(DT.import, startsWith(Period.Desc, "September ")  
 | startsWith(Period.Desc, "October ")  
 | startsWith(Period.Desc, "November ")   
 | startsWith(Period.Desc, "December "))))

## Classification Year Period Period.Desc Aggregate.Level  
## 1: HS 2010 201009 September 2010 6  
## 2: HS 2010 201011 November 2010 6  
## 3: HS 2011 201110 October 2011 6  
## 4: HS 2011 201111 November 2011 6  
## 5: HS 2011 201112 December 2011 6  
## ---   
## 130: HS 2018 201809 September 2018 6  
## 131: HS 2018 201810 October 2018 6  
## 132: HS 2018 201809 September 2018 6  
## 133: HS 2018 201809 September 2018 6  
## 134: HS 2018 201810 October 2018 6  
## Is.Leaf.Code Trade.Flow.Code Trade.Flow Reporter.Code  
## 1: 1 1 Imports 268  
## 2: 1 1 Imports 842  
## 3: 1 1 Imports 51  
## 4: 1 1 Imports 51  
## 5: 1 1 Imports 51  
## ---   
## 130: 1 1 Imports 268  
## 131: 1 1 Imports 268  
## 132: 1 1 Imports 417  
## 133: 1 1 Imports 842  
## 134: 1 1 Imports 842  
## Reporter Partner.Code Partner  
## 1: Georgia 643 Russian Federation  
## 2: United States of America 643 Russian Federation  
## 3: Armenia 643 Russian Federation  
## 4: Armenia 643 Russian Federation  
## 5: Armenia 643 Russian Federation  
## ---   
## 130: Georgia 643 Russian Federation  
## 131: Georgia 643 Russian Federation  
## 132: Kyrgyzstan 643 Russian Federation  
## 133: United States of America 643 Russian Federation  
## 134: United States of America 643 Russian Federation  
## Commodity.Code Commodity Qty.Unit.Code  
## 1: 40510 Dairy produce; derived from milk, butter 0  
## 2: 40510 Dairy produce; derived from milk, butter 0  
## 3: 40510 Dairy produce; derived from milk, butter 0  
## 4: 40510 Dairy produce; derived from milk, butter 0  
## 5: 40510 Dairy produce; derived from milk, butter 0  
## ---   
## 130: 40510 Dairy produce; derived from milk, butter 0  
## 131: 40510 Dairy produce; derived from milk, butter 0  
## 132: 40510 Dairy produce; derived from milk, butter 0  
## 133: 40510 Dairy produce; derived from milk, butter 0  
## 134: 40510 Dairy produce; derived from milk, butter 0  
## Netweight.kg Trade.Value.USD Flag  
## 1: 10000 55926 0  
## 2: NA 4467 0  
## 3: 4190 23740 0  
## 4: 240 1440 0  
## 5: 3942 23652 0  
## ---   
## 130: 518 3203 0  
## 131: 674 3731 0  
## 132: 1465 5087 0  
## 133: 436 4228 0  
## 134: 634 6228 0

#проверим количество пропусков  
na.num <- apply(inform, 2, function(x) length(which(is.na(x))))   
sort(na.num[na.num > 0], decreasing = T) #получили один пропуск

## Netweight.kg   
## 1

# явное преобразование типа, чтобы избежать проблем   
# при заполнении пропусков  
inform[, Netweight.kg := as.double(Netweight.kg) ]  
# считаем медианы и округляем до целого, как исходные данные  
inform[, round(median(.SD$Netweight.kg, na.rm = T), 0),  
 by = Year]

## Year V1  
## 1: 2010 10000  
## 2: 2011 4795  
## 3: 2012 24392  
## 4: 2013 19600  
## 5: 2014 53800  
## 6: 2015 40957  
## 7: 2016 2718  
## 8: 2017 1254  
## 9: 2018 1465

# сначала копируем все значения  
inform[, Netweight.kg.median := round(median(.SD$Netweight.kg,  
 na.rm = T), 0), by = Year]   
  
# затем заменяем пропуски на медианы  
inform[!is.na(Netweight.kg), Netweight.kg.median := Netweight.kg]   
  
# смотрим результат  
inform[, Netweight.kg, Netweight.kg.median]

## Netweight.kg.median Netweight.kg  
## 1: 10000 10000  
## 2: 10000 NA  
## 3: 4190 4190  
## 4: 240 240  
## 5: 3942 3942  
## ---   
## 130: 1465 1465  
## 131: 1577 1577  
## 132: 674 674  
## 133: 436 436  
## 134: 634 634

inform[is.na(Netweight.kg), Year, Netweight.kg.median]

## Netweight.kg.median Year  
## 1: 10000 2010

Разделим данные по странам-членам того или иного союза (получится 3 таблицы). После этого добавим в эти таблички новый столбец - принадлежность к союзу, после чего объединим все таблицы в одну.

#Разделим страны по группам  
unique(DT.import$Reporter)

## [1] "Armenia" "EU-27"   
## [3] "Finland" "Georgia"   
## [5] "Germany" "United States of America"  
## [7] "Belarus" "Estonia"   
## [9] "Kyrgyzstan" "Kazakhstan"   
## [11] "Ukraine" "Azerbaijan"   
## [13] "Lithuania" "Russian Federation"   
## [15] "Latvia" "Mongolia"   
## [17] "New Zealand" "United Arab Emirates"   
## [19] "Slovenia" "Egypt"

#страны таможенного союза  
customs\_union <- c('Armenia',"Belarus", "Kazakhstan", "Kyrgyzstan", "Russian Federation")  
#страны СНГ, не входящий в таможенный союз  
cis <- c("Azerbaijan")  
#остальные страны  
other\_countries <- c("EU-27","Finland","Georgia","Germany","United States of America", "Estonia","Ukraine","Lithuania","Latvia","Mongolia","New Zealand",  
 "United Arab Emirates","Slovenia","Egypt")  
cls <- palette(rainbow(3))# обозначили цвета для графиков  
  
#сделаем поле Reporter ключевым, чтобы сделать возможной фильтрацию  
setkey(inform, Reporter)  
#inform.customs\_union <- inform[c('Armenia',"Belarus",   
# "Kazakhstan", "Kyrgyzstan", "Russian Federation")]  
  
#формируем таблицы по признаку принадлежности стран к тому или иному союзу  
inform.customs\_union <- inform[customs\_union]  
inform.cis <- inform[cis]  
inform.other\_countries <- inform[other\_countries]  
  
#добавим столбец фактор принадлежности к союзу  
inform1 <- mutate(inform.customs\_union, c\_fact='customs\_union')  
inform2 <- mutate(inform.cis, c\_fact='cis')  
inform3 <- mutate(inform.other\_countries, c\_fact='other\_countries')  
  
#объединяем таблицы в одну новую  
inform\_super <- data.table()  
inform\_super <- full\_join(inform1,inform2)

## Joining, by = c("Classification", "Year", "Period", "Period.Desc", "Aggregate.Level", "Is.Leaf.Code", "Trade.Flow.Code", "Trade.Flow", "Reporter.Code", "Reporter", "Partner.Code", "Partner", "Commodity.Code", "Commodity", "Qty.Unit.Code", "Netweight.kg", "Trade.Value.USD", "Flag", "Netweight.kg.median", "c\_fact")

inform\_super <- full\_join(inform\_super, inform3)

## Joining, by = c("Classification", "Year", "Period", "Period.Desc", "Aggregate.Level", "Is.Leaf.Code", "Trade.Flow.Code", "Trade.Flow", "Reporter.Code", "Reporter", "Partner.Code", "Partner", "Commodity.Code", "Commodity", "Qty.Unit.Code", "Netweight.kg", "Trade.Value.USD", "Flag", "Netweight.kg.median", "c\_fact")

В новообразованной таблице превращаем столбец принадлежности к союзу в фактор. после считаем суммарные поставки по союзам и годам, строим график

year\_in <- as.factor(unique(inform\_super$c\_fact))  
  
#считаем суммарные постаки по годам и союзу  
res <- select(inform\_super, Netweight.kg.median, c\_fact,Year) %>%  
 group\_by(c\_fact, Year) %>%  
 mutate(Netweight.kg.total = sum(Netweight.kg.median))  
res1 <- na.omit(res)  
  
#график, построенный с помощью пакета base------   
png(filename = 'Pic-1.png', width = 500, height = 500)  
plot(x = res1$Year, y = res1$Netweight.kg.total, type = 'p',  
 pch = 21, bg = cls[as.factor(res1$c\_fact)],  
 axes = F, ylim=c(0, 1000000),  
 xlim=c(2010,2018),  
 xlab = 'Год продажи',  
 ylab = 'Количество проданной продукции')  
# легенда  
legend('topright', legend = year\_in, fill = cls[year\_in])  
  
# горизонтальная ось  
axis(side = 1, pos = 0, at = seq(2010, 2018, by = 1),  
 labels = seq(2010, 2018, by = 1))  
  
dev.off()

## png   
## 2

Построим оставшиеся 2 графика с помощью lattice и ggplot.

#график, построенный с помощью пакета lattice-----  
png(filename = 'Pic-2.png', width = 500, height = 500)  
xyplot(Netweight.kg.total~Year, data=res1,  
 fill.color = cls[as.factor(res1$c\_fact)],  
 xlab = "Год продажи",  
 ylab = 'Количество проданной продукции',  
 panel = function(x,y,fill.color,...,subscipts){  
 fill = fill.color[subscipts]  
 panel.xyplot(x,y,pch = 19,  
 col = fill)})  
dev.off()

## png   
## 2

#график построенный с помощью ggplot2-----------  
png(filename = 'Pic-3.png', width = 500, height = 500)  
qplot(Year, Netweight.kg.total, data = res1, color = as.factor(c\_fact),  
 xlab = 'Год продажи',  
 ylab = 'Количество проданной продукции' )  
dev.off()

## png   
## 2